Decomposed Meta Batch Normalization for Fast Domain Adaptation in Face Recognition

Jianzhu Guo, Xiangyu Zhu, Zhen Lei, *Senior Member, IEEE*, and Stan Z. Li, *Fellow, IEEE*

***Abstract*—Face recognition systems are sometimes deployed to a target domain with limited unlabeled samples available. For instance, a model trained on the large-scale webfaces maybe required to adapt to a NIR-VIS scenario via very limited unlabeled faces. This situation poses a great challenge to Unsupervised Domain Adaptation with Limited samples for Face Recognition (UDAL-FR), which is less studied in previous works. In this paper, with deep learning methods, we propose a novel training remedy by decomposing the model into the weight parameters and the BN statistics in the training phase. Based on decomposing, we design a novel framework via meta-learning, called *Decomposed Meta Batch Normalization* (DMBN) for fast domain adaptation in face recognition. DMBN trains the network such that domain-invariant information is prone to store in the weight parameters and domain-specific knowledge tends to be represented by the BN statistics. Specifically, DMBN constructs distribution-shifted tasks via domain-aware sampling, on which several meta-gradients are obtained by optimizing discriminative representations across different BNs. Finally, the weight parameters are updated with these meta-gradients for better consistency across different BNs. With the learned weight parameters, the adaptation is very fast since only the BN updating on limited data is needed. We propose two UDAL-FR benchmarks to evaluate the domain-adaptive ability of a model with limited unlabeled samples. Extensive experiments validate the efficacy of our proposed DMBN.**

***Index Terms*—Face Recognition, Unsupervised Domain Adaptation, Meta-learning, Batch Normalization**

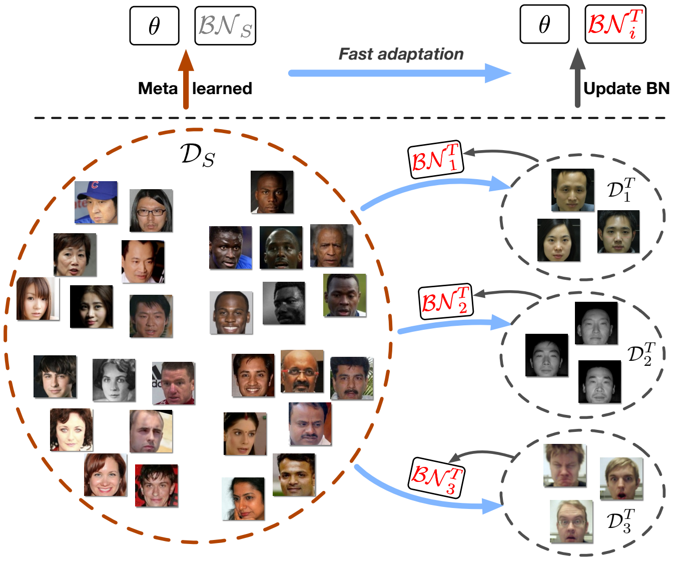


Fig. 1. An illustration of our DMBN for UDAL-FR. The model trained on source domains $\mathcal{D}_S$ is required to fast adapt to a new target domain $\mathcal{D}_T$ with limited unlabeled samples. $\mathcal{D}_1^T$, $\mathcal{D}_2^T$ and $\mathcal{D}_3^T$ in the figure indicate three new target domains, respectively. By meta-learning the weight parameters θ on source domains $\mathcal{D}_S$ with DMBN, our model only needs updating BN statistics $\mathcal{BN}_i^T$ with limited unlabeled samples from the target domain $\mathcal{D}_i^T$ to perform the adaptation, which is very fast.

I. Introduction

FACE recognition has been widely applied in real-world scenarios. Recent works [1], [2], [3], [4], [5], [6] have achieved remarkable performances on common benchmarks, *e.g.*, LFW [7], YTF [8], IJB-C [9] and MegaFace [10], thanks to the deep learning advances based on large-scale training datasets like CASIA-Webface [11], VGGFace2 [12] and MS-Celeb-1M [13]. These methods rely on the underlying assumption that the training and testing sets share similar data distributions. However, in the deployment of face recognition, the trained model sometimes faces a new scenario and is required to adapt to it with limited unlabeled samples available. Such a situation poses a great challenge to the problem of Unsupervised Domain Adaptation with Limited samples for Face Recognition (UDAL-FR), illustrated in Fig. 1.

In comparison with Unsupervised Domain Adaptation for Face Recognition (UDA-FR), UDAL-FR additionally constrains the number of samples from the target domain and is thus more challenging. Among recent works on UDA-FR, one line of works [14], [15] aims at minimizing the discrepancy between the source and target domains, where the domain discrepancy is measured by Maximum Mean Discrepancy (MMD) [16]. Another line of works [17], [18] tries to assign pseudo-labels to samples from the target domain for fine-tuning. Despite the effectiveness on UDA-FR, these methods rely on a large number of samples from the target domain, thus being inappropriate for the UDAL-FR problem.

In this paper, we propose to address the UDAL-FR problem. Once the model is trained, it can fast adapt to a new target domain with limited unlabeled samples. Inspired by AdaBN [19] and MAML [20], we propose a meta-learning based framework by decomposing the model into the weight parameters θ and the BN statistics $\mathcal{BN}$, called *Decomposed Meta Batch Normalization* (DMBN). Based on decomposing, the network is trained such that domain-invariant information is prone to store in θ and domain-specific knowledge tends to be represented by $\mathcal{BN}$. DMBN first constructs a batch of distribution-shifted tasks via domain-aware sampling. Each task consists of two meta batches with distribution shift: meta-train and meta-test batches. Then, DMBN decomposes

J. Guo, X. Zhu, Z. Lei and S. Z. Li are with the Center for Biometrics and Security Research (CBSR), National Laboratory of Pattern Recognition (NLPR), Institute of Automation Chinese Academy of Sciences (CASIA) and University of Chinese Academy of Sciences (UCAS), Beijing, China (e-mail: {jianzhu.guo, xiangyu.zhu, zlei, szli}@nlpr.ia.ac.cn).

the weight parameters θ and the BN statistics $\mathcal{BN}$ for each distribution-shifted task. Finally, we conduct the optimization on these tasks to learn discriminative representations across BNs. The back-propagated meta-gradients from both meta-train and meta-test batches are aggregated to update θ to improve its domain-adaptive ability. In the deployment phase, we only need to update $\mathcal{BN}$ with limited unlabeled samples, which can be very fast.

Compared to traditional meta-learning methods, DMBN is BN agnostic and performs adaptation without gradient updating of the weight parameters. DMBN also outperforms AdaBN [19], which optimizes all the parameters in a regular way during training, leading to the coupling of parameters and BN. The main contributions include: (i) We propose a training strategy by decomposing the network into weight parameters and BN statistics to address the UDAL-FR problem. (ii) A novel meta-learning based optimization framework accompanied with the decomposition strategy, called decomposed meta batch normalization (DMBN) is proposed, so that the learned model is able to adapt to a new target domain efficiently. (iii) To evaluate the performance of different methods on the UDAL-FR problem, two benchmarks are designed and constructed. Extensive experiments on these benchmarks validate the effectiveness of DMBN.

The remainder of this paper is organized as follows. Section II reviews related works. Section III gives a detailed description of our proposed method. Section IV evaluates the efficacy of DMBN by conducting extensive experiments on two proposed benchmarks. We draw a conclusion in Section V.

II. Related work

This section reviews previous works in four aspects: deep face recognition, unsupervised domain adaptation, unsupervised domain adaptation for face recognition and meta-learning.

A. Deep Face Recognition

Since pioneering works DeepFace [1] and DeepID [2], which adopt deep convolutional neural network (CNN) for face recognition, and achieve the performance close to humans on LFW [7] for the first time, then, CNN-based models dominate face recognition. Many powerful loss functions are proposed to learn discriminative representations, so as to improve the performance of deep models. DeepID series [2], [21] use both the identification of softmax loss and the verification of contrastive loss to train the model. FaceNet [3] uses triplet loss to push negative pairs far way from positive pairs by a specific margin and achieves good performance on LFW using 2.6M images. Wen et al. [22] develop a center loss to reduce the intra-class variations. Recently, several margin-based softmax loss functions [23], [4], [24], [5], [25] are proposed to increase the feature margin between different classes. Liu et al. [23] encourage larger inter-class variance by introducing an angular margin between the ground-truth class and other classes (A-Softmax). Liang et al. [26] and Want et al. [25] propose the additive margin (AM-Softmax) to further stabilize the training of A-Softmax. Deng et al. [5] design an additive angular margin (Arc-Softmax) loss with a clear geometric interpretation. Although achieving remarkable performances on standard benchmarks like LFW [7], YTF [8], IJB-C [9] and MegaFace [10], these CNN-based methods rely on a large-scale labeled face set and the distribution between the training and testing sets is similar. If the model is deployed on a target domain with distribution bias, its performance may be degraded dramatically.

B. Unsupervised Domain Adaptation

Unsupervised Domain Adaptation (UDA) aims at transferring knowledge learned from source domains to new domains with unlabeled data only. The main challenge is the domain discrepancy between the source and target domains. In closed-set domain adaptation (DA), many UDA methods are proposed to learn domain-invariant representations with statistic loss [27], [16], [28], [29], [30] or adversarial loss [27], [31], [32], [33], [34], [35]. A commonly used statistic loss for UDA is maximum mean discrepancy (MMD). Deep domain confusion (DDC) [27] simultaneously optimizes the classification loss in the source domain and the MMD metric with an adaptation layer. Deep adaptation network (DAN) [16] utilizes multiple adaptation layers and explores various kernel functions to reduce the shifts in marginal distributions across domains. Adversarial loss is also commonly used to align the distributions of feature space spanned by source and target domains. Domain-adversarial neural network (DANN) [31], [36] introduces a gradient reversal layer to maximize the domain classifier loss and minimize the classification loss adversarially. Li et al. [19] propose a simple strategy named adaptive batch normalization (AdaBN) to perform domain adaptation by only modulating the statistics of BN layers from source domain to the target domain. Closed-set DA assumes that the source and target domains share the same label space. Recently, open-set DA [37], [38], [39], [40] is proposed to extend closed-set DA and has a relax constraint that different domains share partial classes. The key challenge of open-set DA is to separate samples correctly into shared and specific classes. Cao et al. [37] propose a selective adversarial network (SAN) to split the domain discriminator into many class-wise domain discriminators. Zhang et al. [38] propose to identify the importance score of source samples with a two-domain classifier strategy. However, in unsupervised domain adaptation of face recognition, source and target domains do not share the label space, which is a more challenging setting compared to closed-set and open-set DA.

C. Unsupervised Domain Adaptation for Face Recognition

Although deep-learning based methods dominate the recent researches of face recognition, there is only a few studies concentrating on unsupervised domain adaptation. Luo et al. [14] use the Maximum Mean Discrepancy (MMD) loss to decrease domain bias. Sohn et al. [15] synthesize video frames using a series of transformations and utilize still images, synthesized video frames, and unlabeled videos for domain-adversarial training. While achieving promising results, these methods cannot be fast deployed since the computation budget is massive. Moreover, these methods rely on a large number of samples from the target domain, thus being infeasible for the problem of unsupervised domain adaptation with limited samples for face recognition.

D. Meta-learning

The goal of meta-learning is to learn a new task from few samples quickly. Recent studies mainly include three categories: (i) model based [41], [42], (ii) metric-learning based [43], [44], [45], [46] and (iii) optimization based [20], [47], [48], [49] methods. In model based ones, MANN [41] trains a memory-augmented neural network to learn how to store and retrieve memories for each classification task. Munkhdalai et al. [42] propose a meta-learning architecture that learns meta-level knowledge across tasks and changes its inductive bias via fast parametrization. Metric learning based methods focus on learning embeddings that can be recognized with a fixed nearest-neighbor, linear, or CNN classifier. As the foundational work of optimization based methods, MAML [20] learns a good weight initialization for fast adaptation on a new task. The following works Reptile [47], meta-transfer learning [48] and iMAML [49] inherit MAML. Our approach is mostly related to MAML that tries to learn a transferable weight initialization. However, MAML relies on the assumption that new tasks share similar distributions with the training tasks and the tasks are limited to closed-set classification, thus being inappropriate to address the unsupervised domain adaptation problem of open-set face recognition.

III. Methology

This section details the DMBN framework, which aims to address the UDAL-FR problem.

A. Problem Description

For the UDAL-FR problem, we have two datasets: $\mathcal{X}_S$ is the source dataset with several mixed domains $\mathcal{D}_S$ and $\mathcal{X}_T$ is the unlabeled dataset from the target domain $\mathcal{D}_T$. The label sets of $\mathcal{D}_S$ and $\mathcal{D}_T$ are disjoint, and the data distributions between $\mathcal{D}_S$ and $\mathcal{D}_T$ are different. For the target domain $\mathcal{D}_T$, we can only access *a limited number of unlabeled samples*. The goal is to enable the model trained on $\mathcal{D}_S$ to fast adapt to $\mathcal{D}_T$ with limited unlabeled samples.

B. Preliminary of Batch Normalization

We first briefly review Batch Normalization (BN) [50], which is originally proposed to reduce the internal covariate shift by normalizing layer inputs. BN first normalizes each feature independently within a mini-batch and then learns a scale and shift for linear transformation. Formally, given an input $\mathcal{X} \in \mathbb{R}^{N \times C}$, where N is the batch size, the BN serves as a function ϕ_{BN} to transform the input $\mathcal{X}$ into:

$$\begin{aligned} \hat{x}_k &= \frac{x_k - \mathbb{E}\left[\mathcal{X}_{\cdot k}\right]}{\sqrt{\operatorname{Var}\left[\mathcal{X}_{\cdot k}\right] + \epsilon}}, \\ y_k &= \gamma_k \hat{x}_k + \beta_k, \end{aligned} \tag{1}$$

where $k \in \{1, \ldots, n\}$, x_k and y_k are the input and output of the BN layer respectively, $\mathcal{X}_{\cdot k}$ is the k-th column of the input $\mathcal{X}$, and γ_k, β_k are the learnable affine parameters of scale and bias. Note that we use the notation $\mathcal{BN}$ to represent only the mean and variance statistics. The affine parameters are not considered. BN guarantees that the distributions of layers' input are fixed across different mini-batches. In optimization like SGD, fixing the input distribution can greatly accelerate the model convergence [50]. In the conventional testing phase, the BN statistics obtained in training are fixed and used to whiten inputs. Nevertheless, sharing BN statistics for both source and target domains are inappropriate if domain shift exists.

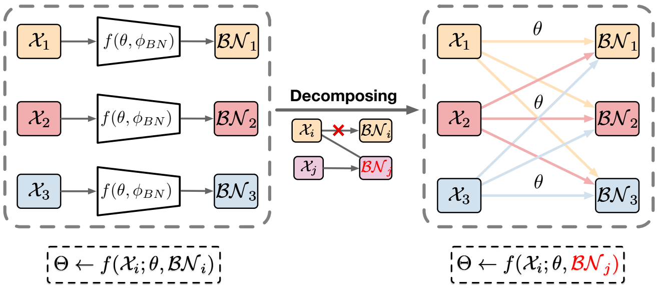


Fig. 2. In vanilla training, the BN statistic $\mathcal{B}_i$ is determined by the input mini-batch $\mathcal{X}_i$, weight parameters θ and the BN function ϕ_{BN}. In our setting, after calculating each $\mathcal{B}_i$, we decompose $\mathcal{X}_i$ and $\mathcal{B}_i$, and then compose the mini-batch $\mathcal{X}_i$ with another $\mathcal{B}_j$. In other words, the decomposing operation cuts off the deterministic relation between $\mathcal{X}_i \xrightarrow{\theta} \mathcal{B}_i$, and compose $\mathcal{B}_j$ derived from another input with $\mathcal{X}_i$: $(\mathcal{X}_i; \theta, \mathcal{BN}_j)$.

C. Decomposing

We decompose the dependency between the weight parameters θ and the BN statistics $\mathcal{BN}$ in the training phase. A model Θ can be defined as a parametrized function with θ and $\mathcal{BN}$: $f(\theta, \mathcal{BN})$. Our goal is learning θ and $\mathcal{BN}$ independently such that domain-invariant information is prone to store in θ and domain-specific information tends to be represented by $\mathcal{BN}$. Given an input mini-batch $\mathcal{X}_i$, the BN statistics $\mathcal{BN}_i$ is determined by θ and $\mathcal{X}_i$: $\mathcal{BN}_i = (\phi_{BN} \circ f)(\mathcal{X}_i; \theta)$, where ϕ_{BN} is the normalization function of BN. In vanilla training, θ and $\mathcal{BN}_i$ are tightly coupled since $\mathcal{BN}_i$ relies on θ. This coupled relation does not meet our requirement. We design to make the weight parameters θ (i) robust across different $\mathcal{BN}$, and (ii) ready for fast adaptation to a target domain via generating $\mathcal{BN}$ with limited samples from the target domain. To achieve this, we propose to decompose θ and $\mathcal{BN}$ in the training phase. Specifically, when fed with a mini-batch $\mathcal{X}_i$, the corresponding $\mathcal{BN}$ is not directly derived from θ and $\mathcal{X}_i$. The decomposing is formally represented by $f(\mathcal{X}_i; \theta, \mathcal{BN}_i) \rightarrow f(\mathcal{X}_i; \theta, \boldsymbol{\mathcal{BN}_j})$, where $\mathcal{BN}_j$ is from another input $\mathcal{X}_j$. The detailed illustration is shown in Fig. 2.

D. Distribution-shifted Task Sampling

Decomposing exactly cuts off the deterministic relation between the input, weight parameters and the BN statistics:

$(\mathcal{X}, \theta) \rightarrow \mathcal{BN}$. To make the weight parameters θ domain-invariant, we construct distribution-shifted tasks by domain-aware sampling to synthesize the distribution shift for training. The overview is shown in Fig. 3. Specifically, in each training iteration, we first sample a batch of tasks $\{\mathcal{T}_i | i = 1, 2, 3 \cdots\}$ from source mixed domains $\mathcal{D}_S$. Each task $\mathcal{T}_i$ consists of a meta-train batch $\mathcal{X}^i_{mtr}$ a meta-test batch $\mathcal{X}^i_{mte}$, sampled from the same domain $\mathcal{D}_i \in \mathcal{D}_S$. For each meta-train or meta-test batch, we randomly sample B individuals. For each individual, we randomly sample two face images, in which one as the gallery image and another one as the probe. Note that $\mathcal{X}^i_{mtr}$ and $\mathcal{X}^i_{mte}$ are without overlapped individuals. Since the meta-train and meta-test batches are sampled from the same domain, the distribution shift between them is limited. To enlarge the gap, we randomly shuffle the meta-train batches across different tasks. By doing so, the meta-train batch $\mathcal{X}^i_{mtr}$ of the domain $\mathcal{D}_i$ corresponds to a random meta-test batch $\mathcal{X}^j_{mte}$ of $\mathcal{D}_j$, shown in Fig. 3 (c). Once a batch of distribution-shifted tasks are built, we decompose the deterministic relation between the meta-train batch and its BN statistics: $\mathcal{X}^i_{mtr} \rightarrow \mathcal{B}^i_{mtr}$, and then compose the weight parameters, BN statistics and the input: $\{\mathcal{X}^i_{mte}; \theta, \mathcal{B}^j_{mtr}\}$.

E. Learning Representation Across BNs by Hard-pair Loss

To learn discriminative representations/features for each task, we adapt the hard-pair attention loss in MFR [51] to fit our setting. The hard-pair attention loss focuses on optimizing hard positive and negative pairs to enforce the feature (the activations of the last global average pooling layer) more discriminative. Since we decompose the weight parameters θ and the BN statistics $\mathcal{BN}$, the face features of the meta-test batch $\mathcal{X}_{mte}$ cannot be extracted directly. We first calculate $\mathcal{BN}_{mtr} = (\phi_{BN} \circ f)(\mathcal{X}_{mtr}; \theta)$ using the meta-train batch $\mathcal{X}_{mtr}$, then extract the gallery and probe feature of the meta-test batch: $F^{mte}_g = f(\mathcal{X}^{mte}_g; \theta, \mathcal{BN}_{mtr}) \in R^{B \times C}$, $F^{mte}_p = f(\mathcal{X}^{mte}_p; \theta, \mathcal{BN}_{mtr}) \in R^{B \times C}$, where C is the dimension of the extracted feature. l_2 normalization is performed on each row of the features. The similarity matrix of the meta-test batch is next constructed by $M_{mte} = F^{mte}_g F^{mte^T}_p$. For the meta-train batch, the similarity matrix M_{mtr} is constructed regularly. Different from [51], we use two proportion factors δ_p and δ_n to filter hard positive and negative pairs. δ_p and δ_n can directly reflect the degree of difficulty of positive and negative pairs, and also guarantee the number of hard pairs is balanced. Then, we sort the positive and negative pairs by the similarity score and optimize the hardest pairs. The loss on the meta-train or meta-test batch is thus formulated as:

$$\mathcal{L}_{hp} = \frac{1}{2|\mathcal{P}|} \sum_{i \in \mathcal{P}} \|F_{g_i} - F_{p_i}\|_2^2 - \frac{1}{2|\mathcal{N}|} \sum_{(i,j) \in \mathcal{N}} \|F_{g_i} - F_{p_j}\|_2^2, \tag{2}$$

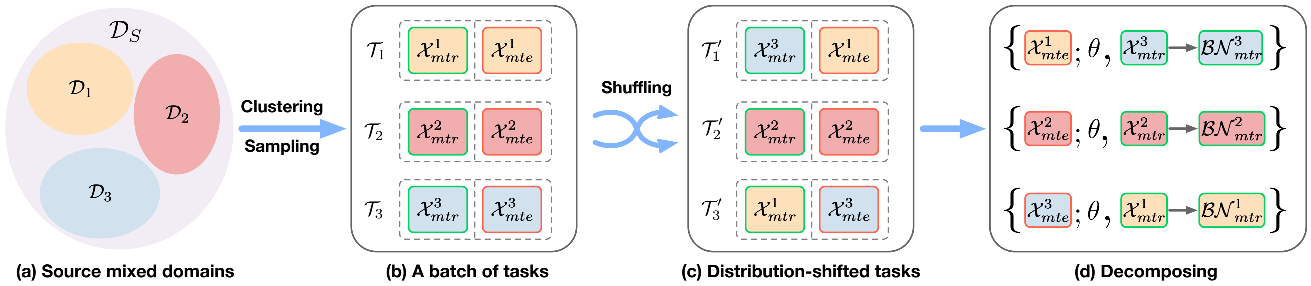
where $\mathcal{P}$ is the indices of top hardest $\delta_p \cdot B$ positive pairs and $\mathcal{N}$ is the indices of top hardest $\delta_n \cdot (B^2 - B)$ negative pairs.

Algorithm 1: The algorithm overview of balanced k-means.

```
Function BalancedKMeans ($F$, $k$) :
Let $\{c_i | i = 1, \cdots, k\}$ be initialized cluster centers by
  $k$-means++ [52];
// Initialization
Let $N$ be the total number of features in $F$ and $M$ be the
  desired size of each cluster: $M = N/k$;
Sort $\{F_i | i = 1, \cdots, N\}$ by the absolute value of the
  distance to the farthest center minus the distance to the
  closest candidate center in descending order:
  $\big| \|F_i - c_j\|_2 - \|F_i - c_l\|_2 \big|$, where
  $j = \arg\max_j \|F_i - c_j\|_2$, $l = \arg\min_l \|F_i - c_l\|_2$;
for each $F_i \in F$ do
    If cluster $c_l$ is not full, assigning $F_i$ to cluster $c_l$;
    Otherwise, assigning $F_i$ to the first not-full cluster sorted
      by the absolute difference;
end
// Adjustment
Update centers by the current assignment;
while ite $\leq$ max_iterations do
    Re-calculate the assignment by current centers;
    Sort $\{F_i | i = 1, \cdots, N\}$ by the absolute difference
      between the distance to the current and best assigned
      center in descending order;
    Initialize an empty list: candidate;
    for each $F_i \in F$ do
        Let moved_flag be false;
        for $F_j \in$ candidate do
            if exchange the center assignment between $F_i$
              and $F_j$ gives smaller inertia then
                Exchange and set moved_flag true, break;
            end
        end
        Append $F_j$ to candidate if moved_flag is false;
    end
    Update centers by the current assignment;
end
Output: $k$ balanced domains
```

F. Domain Labels by Balanced k-means

Distribution-shifted task sampling in Section III-D relies on domain labels of training datasets. However, in many real-world applications, the domain label is not always available or is difficult to label. As we know, the domain is affected by many factors like age, race, expression, external environmental variations [53] etc. To address this problem, we propose to divide domains based on the similarity of visual features automatically. A straightforward method is k-means. However, in vanilla k-means, the size of each cluster is random, which is harmful to the hard pair mining. In training, the hard-pair loss (see Section III-E) relies on the size B and two difficulty factors δ_p and δ_n to filter and balance the hard pairs. For each cluster with size B, $\delta_n \cdot (B^2 - B)$ negative pairs and $\delta_p \cdot B$ positive pairs are selected out for training. The randomness of size B brought by vanilla k-means will make the number of hard pairs inconsistent across different clusters, thus making the optimization unstable. To alleviate it, we improve the vanilla k-means by ensuring cluster size to be similar in the whole clustering procedure. We name it balanced k-means. The core of the balanced k-means includes: (i) *Initialization*. Initialize centers by k-means++ [52], sort the features by the absolute value of the distance to their farthest center *minus* the distance to their closest candidate center in descending order, and then assign each feature to its closest candidate center/cluster until the cluster is full. (ii)


$\mathcal{D}_S$
$\mathcal{D}_1$
$\mathcal{D}_2$
$\mathcal{D}_3$
Clustering
Sampling
$\mathcal{T}_1$ $\mathcal{X}^1_{mtr}$ $\mathcal{X}^1_{mte}$
$\mathcal{T}_2$ $\mathcal{X}^2_{mtr}$ $\mathcal{X}^2_{mte}$
$\mathcal{T}_3$ $\mathcal{X}^3_{mtr}$ $\mathcal{X}^3_{mte}$
Shuffling
$\mathcal{T}'_1$ $\mathcal{X}^3_{mtr}$ $\mathcal{X}^1_{mte}$
$\mathcal{T}'_2$ $\mathcal{X}^2_{mtr}$ $\mathcal{X}^2_{mte}$
$\mathcal{T}'_3$ $\mathcal{X}^1_{mtr}$ $\mathcal{X}^3_{mte}$
$\{\mathcal{X}^1_{mte}; \theta, \mathcal{X}^3_{mtr} \rightarrow \mathcal{BN}^3_{mtr}\}$
$\{\mathcal{X}^2_{mte}; \theta, \mathcal{X}^2_{mtr} \rightarrow \mathcal{BN}^2_{mtr}\}$
$\{\mathcal{X}^3_{mte}; \theta, \mathcal{X}^1_{mtr} \rightarrow \mathcal{BN}^1_{mtr}\}$

(a) Source mixed domains (b) A batch of tasks (c) Distribution-shifted tasks (d) Decomposing

Fig. 3. The overview of distribution-shifted task sampling. Given N source mixed domains $\{\mathcal{D}_i | i = 1, 2, 3, \cdots\}$, we first sample a batch of tasks $\{\mathcal{T}_i | i = 1, 2, 3, \cdots\}$, and each task consists of a meta-train $\mathcal{X}^i_{mtr}$ and meta-test batch $\mathcal{X}^i_{mte}$ from the same domain (or pseudo-domain generated by balanced k-means). We then randomly shuffle the meta-train batches across tasks to enlarge the distribution bias between meta-train and meta-test. Note that after shuffling, the meta-train and meta-test batches may be still from the same domain (*e.g.*, the second task $\mathcal{T}_2$ in the figure). Finally, we decompose the weight parameters θ and BN statistics $\mathcal{BN}$ for training.

Adjustment. During each iteration, the feature prefers to be exchanged if the absolute value of the distance to its current assigned center *minus* the distance to its best candidate center is large. Specifically, we sort the features based on the absolute difference value in descending order, then exchange two features if the exchange brings improvement, while keeping the same cluster size via maintaining a candidate list. The detailed algorithm is described in Algorithm 1. Balanced k-means is directly incorporated into the DMBN framework to assign pseudo-domain labels within each batch. Note that the objective of balanced k-means is to cluster the samples into visually discriminative groups evenly for building distribution-shifted tasks, not to predict the specific domain labels. In several unbalanced conditions, the domain labels by balanced k-means may not correspond to some manually defined ones (*e.g.*, race labels in source domains), but may be clustered by other semantic attributes. For example, Caucasians may dominate the dataset distribution, but the Caucasian faces can be further clustered into different groups by the pose or age attribute, which is sound for constructing distribution-shifted tasks.

G. Meta-optimization

This section details the optimization procedure to improve the domain-adaptive ability of the model. The whole optimization procedure can be referred in Algorithm 2.

Meta-train. In each task, the meta-train batch $\mathcal{X}_{mtr}$ contains B paired images $\mathcal{X}_{mtr}$. We conduct the adapted hard-pair attention loss as follows:

$$\mathcal{L}_{mtr} = \mathcal{L}_{hp}(\mathcal{X}_{mtr}; \theta, \mathcal{BN}_{mtr}), \tag{3}$$

where θ is the weight model parameters, and the BN statistics $\mathcal{BN}_{mtr}$ is derived with θ and $\mathcal{X}_{mtr}$. The back-propagated gradient is denoted as $\nabla_\theta \mathcal{L}_{mtr}$. This step is similar to the conventional training since $\mathcal{BN}_{mtr}$ corresponds to the input $\mathcal{X}_{mtr}$.

Meta-test. In each task, the model is also tested on the meta-test batch, which has a distribution shift with the meta-train batch. It simulates the real-world adaptation so as to make the learned weight parameters more adaptive to an unknown domain. Specifically, the meta-train batch with limited samples represents the local distribution of a domain, while the meta-test simulates the unseen distribution to be evaluated. Since we decompose the weight parameters θ and the BN statistics $\mathcal{BN}_{mte}$, the meta-test loss is conducted with another $\mathcal{BN}_{mtr}$:

$$\mathcal{L}_{mte} = \mathcal{L}_{hp}(\mathcal{X}_{mte}; \theta, \mathcal{BN}_{mtr}), \tag{4}$$

Overall objective. To combine the optimization on the meta-train and meta-test, we build the final objective as:

$$\arg\min_{\theta} \gamma \mathcal{L}_{mtr}(\theta) + (1 - \gamma)\mathcal{L}_{mte}(\theta), \tag{5}$$

where γ weights the meta-train and meta-test losses. This objective can be understood as: *optimize the weight parameters θ, not only to fit the meta-train domain, but also learn to fast adapt to the distribution-shifted meta-test domain with limited samples.* From another view, the second term of Eqn. 5 can be regarded as an extra regularization to encourage θ more robust across domains. Finally, the learned weight parameters θ can well adapt to a target domain, via only updating BN with limited samples. The overview of DMBN is described in Algorithm 2.

H. Fast Adaptation to Target Domain

To adapt to a new target domain, the model trained by DMBN only needs to update the BN statistics with limited unlabeled samples from the target domain. In the adaptation phase, given a batch of m unlabeled samples, the global mean μ_i and variance σ_i^2 of the BN statistics can be estimated as follows:

$$\begin{aligned} n_i &= n_{i-1} + m, \\ \delta &= \hat{\mu} - \mu_{i-1}, \\ \mu_i &\leftarrow \frac{n_{i-1} \cdot \mu_{i-1} + m \cdot \hat{\mu}}{n_i}, \\ \sigma_i^2 &\leftarrow \frac{\hat{\sigma}^2 + \sigma_{i-1}^2 + \delta^2 \cdot n_{i-1} \cdot m}{n_i}, \end{aligned} \tag{6}$$

where $\hat{u}$ and $\hat{\sigma}^2$ are the mean and variance estimation of the current input mini-batch, n_i is the aggregated number of samples from the past iterations. Note that when $i = 0$, n_i, μ_i and σ_i^2 are initialized to 0, $\mathbf{0}$ and $\mathbf{1}$, respectively. The estimation follows the online parallel algorithm proposed

Algorithm 2: The algorithm overview of DMBN.

```
Input: Source mixed domains: $\mathcal{D}_S$.
Init: A pre-trained model $f(\theta)$ with the weight parameters
      $\theta$, the meta-train and meta-test batch-size of $B$, and
      hyper-parameters $\beta$, $\gamma$.
for ite in max_iterations do
    Init the gradient $g_\theta$ as $\mathbf{0}$;
    // Sample a batch of tasks
    if the domain labels are known then
        Init $N$ as the number of source domains $\mathcal{D}_S$;
        for each $\mathcal{D}_i \in \mathcal{D}_S$ do
            // Sampling a task
            Sample B paired images from B individuals of
              $\mathcal{D}_i$ for meta-train batch $\mathcal{X}^i_{mtr}$;
            Sample B paired images from other B
              individuals of $\mathcal{D}_i$ for meta-test batch $\mathcal{X}^i_{mte}$;
        end
    else
        Init $N$ as the cluster number $k$ of balanced $k$-means;
        Sample $2k \cdot B$ samples $\mathcal{X}$ from $\mathcal{D}_S$ and extract
          features $F$;
        Split $\mathcal{X}$ by domain labels returned from
          BalancedKMeans ($F$, $k$) into meta-train $\mathcal{X}^i_{mtr}$ and
          meta-test batches $\mathcal{X}^i_{mte}$;
    end
    // Construct distribution-shifted tasks
    Shuffle the indices of the sampled meta-train batches:
      $y = shuffle(\{1, 2, \cdots, N\})$;
    for $i = 1 \rightarrow N$ do
        Calculate BN statistics of each meta-train batch:
          $\mathcal{BN}^i_{mtr} = (\phi_{BN} \circ f)(\mathcal{X}^i_{mtr}; \theta)$;
    end
    // Decomposing and composing
    for $i = 1 \rightarrow N$ do
        Meta-train:
        $\mathcal{L}_{mtr} = \mathcal{L}_{hp}(\mathcal{X}^i_{mtr}; \theta, \mathcal{BN}^i_{mtr})$;
        Meta-test:
        $j = y(i)$;
        $\mathcal{L}_{mte} = \mathcal{L}_{hp}(\mathcal{X}^i_{mte}; \theta, \boldsymbol{\mathcal{BN}^j_{mtr}})$;
        Gradient aggregation:
        $g_\theta \leftarrow g_\theta + \gamma \nabla_\theta \mathcal{L}_{mtr} + (1-\gamma) \nabla_\theta \mathcal{L}_{mte}$;
    end
    Meta-optimization:
    Update $\theta \leftarrow \theta - \frac{\beta}{N} g_\theta$ by $SGD$;
end
```

in [54]. This implementation is numerically stable and can fast estimate the global mean and variance with limited GPU memory.

IV. Experiments

To demonstrate the efficacy of DMBN, we design two UDAL-FR benchmarks for evaluation and conduct several experiments on the proposed benchmarks.

A. UDAL-FR Benchmarks and Protocols

The first UDAL-FR-I benchmark assumes that domain labels of source mixed domains are accessible, while the second UDAL-FR-II benchmark assumes not. The setting of UDAL-FR-II is more challenging than UDAL-FR-I.

Training datasets. In real-world scenarios, a large base dataset is usually used to pre-train a model. The pre-trained model may generalize poorly on target scenarios, thus being required to perform adaptation. To build such a benchmark, we use Ms-Celeb-1M-NR as the base dataset and RFW [55] as our source training domains following [51]. Ms-Celeb-1M-NR indicates Ms-Celeb-1M without RFW, since RFW overlaps Ms-Celeb-1M [13] with a few individuals. In Ms-Celeb-1M-NR, the overlapped individuals with RFW are removed according to the individual keyword. Ms-Celeb-1M-NR is thus independent of source training domains. Specifically, the source training domain **RFW** [55] consists of four subsets, namely Caucasian, Asian, African and Indian. For each subset, we select about 2K individuals as a source domain for training following [51]. Detailed statistics of the source training domain are shown in Table I.

Testing datasets. **CASIA NIR-VIS 2.0** [56] is a large and challenging face dataset across NIR and VIS spectrum. We follow the standard protocol defined by [56] to use 6,566 images of 358 individuals for testing. 8,749 samples of the training set are made unlabeled for the sampling and fast adaptation. Note that there are 10 folds and we only show the number of individuals and images of the first fold in Table II. Other folds have similar statistics and we report the average value of 10 folds. **HFB** [57] is an older but also widely used dataset across NIR and VIS spectrum. Following [58], we use 2,918 images of 102 subjects for testing. 2,157 samples of the training set are made unlabeled for the sampling and fast adaptation. **Oulu-CASIA NIR-VIS** [59] consists of 80 subjects with six expression variations (anger, disgust, fear, happiness, sadness and surprise). Following [60], we adopt 1,920 images of 20 subjects for testing. 5,760 samples of the training set are made unlabeled for the sampling and fast adaptation. **MultiPIE** is adopted for cross-pose evaluation following [51]. 1,690 images of 237 individuals are selected for testing. 18,704 unlabeled samples without overlapped individuals are for the sampling and fast adaptation. **MeGlass** [61] is used for evaluating eyeglass-robust face recognition. We split the original 1,710 individuals into two parts: 6,040 images of 1,510 individuals for testing, and the remaining 19,660 unlabeled samples of 200 individuals for the sampling and fast adaptation. **Public-IvS** [62] considers the ID vs. Spot face recognition. We select 4,241 images of 1,012 individuals for testing and the remaining 1,159 unlabeled samples of 200 individuals for the sampling and fast adaptation. **WebCaricature** [63] is originally proposed for caricature face recognition. We follow the unrestricted face verification protocol in [63] for testing. Specifically, 1,094 images of 26 individuals are for testing and 10,924 unlabeled samples of 226 individuals are for the sampling and fast adaptation. The testing protocols of CASIA NIR-VIS 2.0, HFB, Oulu-CASIA NIR-VIS and WebCaricature are the same as the original ones. The testing protocols of MultiPIE, MeGlass and Public-IvS are slightly different from [51] since a portion of samples from target domains are made unlabeled for fast adaptation. Detailed statistics of testing datasets are shown in Table II.

Benchmark protocol. UDAL-FR-I consists of the source domain dataset RFW with known domain labels for Caucasian, Asian, African and Indian and seven target datasets. While in UDAL-FR-II, the domain information in the source domain dataset is unknown. This is the only difference between UDAl-FR-I and UDAl-FR-II. UDAL-FR-II is more consistent with real-world scenarios and also more challenging than UDAL-FR-I.

Evaluation method. During testing, we extract each face image's feature and its flipped one, then concatenate them to construct the final representation. Cosine similarity is used as the measured score. We use the receiver operating characteristic (ROC) curve and Rank-1 accuracy to evaluate the performance. For ROC, we report the verification rate (VR) at low false acceptance rates (FAR) *e.g.*, 1%, 0.1%, 0.01%, and 0.001%. For Rank-1 accuracy, each probe image is matched to all gallery images. If the top-1 return is the same individual, the matching is correct. For WebCaricature, we only report VRs at FAR=1% and 0.01% following the original verification protocol [63].

Adaptation setting. For each target domain $\mathcal{D}_i^T$, we randomly sample a limited number of samples, *e.g.*, 16, 32, 64 or 1,000 samples from unlabeled images in $\mathcal{D}_i^T$ for the unsupervised domain adaptation. We repeat the sampling ten times independently and report the mean value. In adaptation, the mini-batch size is set to 64. If the number of target samples is less than 64, we directly estimate the BN statistics $\mathcal{BN}$ composing of the mean and variance within a batch. Otherwise, we adopt the online algorithm in Section III-H to estimate $\mathcal{BN}$. Once we acquire $\mathcal{BN}$, we use the weight parameters θ and $\mathcal{BN}$ to extract face representations and perform testing.

TABLE I
THE STATISTICS OF SOURCE DOMAINS $\mathcal{D}_S$ IN UDAL-FR-I AND UDAL-FR-II BENCHMARKS. DM. INDICATES DOMAIN LABEL.

Protocol	Dataset of $\mathcal{D}_S$	#Train ID	#Train img.
UDAL-FR-I	Caucasian	1,957	6,757
	Asian	1,492	5,784
	African	1,995	6,938
	Indian	1,984	6,857
UDAL-FR-II	Mixed w/o dm.	7,428	26,336

TABLE II
THE STATISTICS OF TARGET DOMAINS $\mathcal{D}_T$. WE RE-ORGANIZE THE DATASETS TO ENSURE THE UNLABELED IMAGES AND TESTING IMAGES ARE DISJOINTED WITHOUT OVERLAPPED INDIVIDUALS.

Dataset of $\mathcal{D}_T$	#Unlabeled img.	#Test ID	#Test img.
CASIA NIR-VIS 2.0	8,749	358	6,566
HFB	2,157	102	2,918
Oulu-CASIA NIR-VIS	5,760	20	1,920
MultiPIE	18,704	237	1,690
MeGlass	19,660	1,510	6,040
Public-IvS	1,159	1,012	4,331
WebCaricature	10,924	26	1,094

B. Implementation Details

Our experiments are based on PyTorch [64]. The random seed on CPU and GPU is fixed as 2,020 in all comparative experiments for fair comparisons. We use a 28-layer ResNet as the backbone, with about 129M MACs and 4.6M parameters. The feature/representation dimension is 256. We pre-train our model on Ms-Celeb-1M-NR using CosFace [24]. The input face image is aligned, then cropped and resized to 120×120. The input batches are normalized by subtracting 127.5 and being divided by 128. The optimization step-size β is initialized to 0.0001. The weight γ balancing the meta-train and meta-test losses is set to 0.5. The batch-size B is 64. During training, the step-size β is decayed by 0.5 every 1K steps, and the total iterations are 8K. The whole training time is about 8 hours on a TITAN Xp GPU. δ_p is set to 0.4 and δ_n is 0.01. k is set to 4 for the balanced k-means. In meta-optimization, we choose SGD to optimize the network. The weight decay is 0.0005, and the momentum is 0.9.

C. Comparative Experiments with Different Numbers of Samples and Baselines

Settings. We evaluate our method DMBN with different numbers of unlabeled samples and compare it to two baselines. The numbers contain four options: 16, 32, 64 and 1,000. The baselines include: (i) *Agg*: the model pre-trained on the aggregation of MS-Celeb-1M-NR and source domains using CosFace [24]. We use the *Agg* model as a simple baseline for comparison. (ii) *Base*: the model fine-tuned on source domains by the hard-pair attention loss. We use *Base* as a strong baseline for comparison. We report the VRs at low FAR 1%, 0.1%, 0.01%, 0.001% and the Rank-1 accuracy. Except for the *Agg* and *Base* model, we repeat the experiments ten times independently and report the mean value.

Results. The comparative results are shown in Table III. From the results, we can make the following observations: (i) The baseline models of *Agg* and *Base* are strong in general, but do not perform well at some low FARs. For example, the *Agg* or *Base* model only achieves 47.55% or 66.31% VR on HFB at FAR=0.001% and 47.13% or 55.62% VR on WebCaricature at FAR=1%. This is possibly because the bias between source and target domains is too large. In comparison, our method DMBN surpasses these two baselines by a significant margin among all seven target domains with only a limited number of unlabeled samples. Specifically, our DMBN improves the VR over 10% compared to baselines on the most challenging domain of WebCaricature. (ii) Generally, our method DMBN is robust to different numbers of samples. When the number of target samples reduces from 1,000 to 32 or 16, the performance drop is little, and the performance improvement over *Base* is still apparent. For example, when the number reduces from 1,000 to 16 on CASIA NIR-VIS 2.0, the VR drops less than 1% at FAR=0.01%, but still improves 5.72% over the *Base* model.

D. Comparative Experiments to UDA Competitors with Very Limited Target Samples

Settings. To evaluate the effectiveness of DMBN, we compare DMBN to other competitors of UDA methods with a very limited number of target samples, *e.g.*, 16 samples on all target domains, which is a very challenging setting. The UDA competitors include: (i) *AdaBN (Agg)* and *AdaBN (Base)*: the *Agg* and *Base* models adapted to target domain via

TABLE III

COMPARISON RESULTS WITH DIFFERENT NUMBERS OF UNLABELED SAMPLES AND TWO BASELINES ON UDAL-FR-I. EXCEPT FOR *Agg* AND *Base*, WE REPORT THE MEAN VALUE OF TEN INDEPENDENT EXPERIMENTS. #TARGET SAMPLES INDICATE THE NUMBER OF UNLABELED SAMPLES FROM THE TARGET DOMAIN. WE HIGHLIGHT THE BEST RESULT OF EACH TARGET DATASET/DOMAIN.

Dataset / Domain	Method	#Target samples	VR (%)				Rank1 (%)
			FAR=1%	FAR=0.1%	FAR=0.01%	FAR=0.001%	
CASIA NIR-VIS 2.0	Agg [24]	-	97.55	89.93	70.82	51.8	92.86
	Base [51]	-	98.52	93.36	80.09	58.12	94.51
	DMBN	16	99.21	94.98	85.81	70.6	96.5
		32	99.23	95.1	85.93	70.88	96.61
		64	99.22	94.96	86.01	70.84	96.58
		1,000	**99.36**	**96.19**	**86.8**	**72.11**	**97.12**
HFB	Agg [24]	-	99.33	94.68	78.7	47.55	99.12
	Base [51]	-	99.67	97.25	83.93	66.31	99.45
	DMBN	16	99.96	98.85	95.38	85.12	99.97
		32	99.98	99.08	96.45	**87.87**	99.99
		64	**99.99**	99.07	96.63	87.27	99.99
		1,000	**99.99**	**99.09**	**96.72**	87.76	**100**
Oulu-CASIA NIR-VIS	Agg [24]	-	95.09	82.54	71.02	59.77	100
	Base [51]	-	96.82	86.4	71.63	61.84	100
	DMBN	16	96.32	86.12	73.7	63.35	100
		32	96.5	86.61	74.46	65.82	100
		64	**96.85**	88.01	76.63	68.6	100
		1,000	**96.85**	**88.91**	**76.93**	**69.12**	100
MultiPIE	Agg [24]	-	100	99.89	98.51	90.28	100
	Base [51]	-	100	100	99.15	92.35	100
	DMBN	16	100	100	99.43	**96.63**	100
		32	100	100	99.51	96.35	100
		64	100	100	99.6	96.44	100
		1,000	100	100	**99.61**	96.62	100
MeGlass	Agg [24]	-	98.86	94.9	86.01	72.04	97.91
	Base [51]	-	98.57	95.58	88.06	74.81	98.18
	DMBN	16	99.03	96.15	89.43	79.11	98.43
		32	99.15	96.51	90.28	80	98.6
		64	99.17	**96.52**	90.48	80.29	**98.62**
		1,000	**99.21**	**96.52**	**90.81**	**80.77**	98.6
Public-IvS	Agg [24]	-	97.94	94.2	86.33	74.96	93.37
	Base [51]	-	98.11	94.31	87	76.51	93.74
	DMBN	16	98.41	96.73	92.35	84.71	96
		32	98.39	96.81	92.8	85.63	96.01
		64	98.41	96.83	93.02	**86.13**	96.05
		1,000	**98.42**	**96.9**	**93.15**	85.91	**96.06**
WebCaricature	Agg [24]	-	47.13	34.86	-	-	-
	Base [51]	-	55.62	39.75			
	DMBN	16	66.29	46.13	-	-	-
		32	66.23	47.01			
		64	66.64	47.37			
		1,000	**67.55**	**48.09**			

AdaBN [19]. (ii) *MMD (Base)*: the *Base* model trained with the Maximum Mean Discrepancy (MMD) regularization. We adopt two MMD kernels for comparisons: the linear kernel *MMD-Linear (Base)* and the multiple (five) gaussian kernels *MMD-Gaussian (Base)* following [14]. (iii) *Pseudo labeling + Imp.*: the model trained with pseudo-labeled samples of the target domain. NANN [17] is first adapted to assign pseudo-labels to unlabeled target samples, then the model is fine-tuned by weight-imprinted method [65]. Note that we cannot generate enough hard pairs with limited target samples, thus fine-tuning with the hard-pair attention loss is infeasible.

Results. From the comparative results in Table IV and Table III, we can conclude: (i) Overall, our method DMBN achieves the best results on seven target domains compared to all other UDA competitors. (ii) The improvement of AdaBN is limited for the *Agg* and *Base* models. On HFB, the *AdaBN (Agg)* or *AdaBN (Base)* model even brings negative adaptation when FAR is 0.001%. Generally, AdaBN improves the adaptation performance, but is unstable and seems not very effective with very limited samples. (iii) The multi-gaussian kernels of MMD perform better than the linear kernel, but the performance gain is small on all target domains. (iv) Pseudo labeling based methods perform badly when the target number is 16. It is possibly because fine-tuning on the

TABLE IV
COMPARISON RESULTS WITH OTHER UDA COMPETITORS WITH 16 TARGET SAMPLES, UNDER THE UDAL-FR-I BENCHMARK. WE REPORT THE MEAN VALUE OF TEN INDEPENDENT EXPERIMENTS. #TARGET SAMPLES INDICATE THE NUMBER OF UNLABELED SAMPLES FROM THE TARGET DOMAIN. WE HIGHLIGHT THE BEST RESULT OF EACH TARGET DATASET/DOMAIN EXCEPT FOR 100% VR AND RANK1 ACCURACY.

Dataset / Domain	**Method**	**#Target samples**	**VR (%)**				**Rank1 (%)**
			FAR=1%	**FAR=0.1%**	**FAR=0.01%**	**FAR=0.001%**	
CASIA NIR-VIS 2.0	AdaBN (Agg) [19]	16	97.34	89.56	72.04	47.53	93.11
	AdaBN (Base) [19]		98.24	93	80.16	60.32	95.06
	MMD-Linear (Base) [14]		98.18	92.25	79.75	64.3	93.78
	MMD-Gauissian (Base) [14]		97.95	92.82	80.69	64.04	93.99
	Pseudo labeling [17] + Imp. [65]		97.83	91.3	75.86	52.96	93.27
	DMBN (Ours)		**99.21**	**94.98**	**85.81**	**70.6**	**96.5**
HFB	AdaBN (Agg) [19]	16	99.11	95.6	78.34	39.99	97.98
	AdaBN (Base) [19]		99.45	96.86	84.23	57.95	99.3
	MMD-Linear (Base) [14]		99.55	94.89	81.17	63.1	99.73
	MMD-Gauissian (Base) [14]		99.49	94.36	83.09	63.51	**100**
	Pseudo labeling [17] + Imp. [65]		99.05	96	81.19	48.24	98.9
	DMBN (Ours)		**99.96**	**98.85**	**95.38**	**85.12**	**100**
Oulu-CASIA NIR-VIS	AdaBN (Agg) [19]	16	92.89	78.05	61.96	49.62	99.76
	AdaBN (Base) [19]		96.04	85.27	71.77	61	100
	MMD-Linear (Base) [14]		95.46	84.59	68.07	56.06	100
	MMD-Gauissian (Base) [14]		94.95	83.25	70.02	57.59	100
	Pseudo labeling [17] + Imp. [65]		96.29	85.92	72.96	59.5	100
	DMBN (Ours)		**96.32**	**86.12**	**73.7**	**63.35**	100
MultiPIE	AdaBN (Agg) [19]	16	100	99.88	97.77	90	100
	AdaBN (Base) [19]		100	99.94	98.61	93.59	100
	MMD-Linear (Base) [14]		100	99.97	99.12	94.75	100
	MMD-Gauissian (Base) [14]		100	99.86	98.57	94.17	100
	Pseudo labeling [17] + Imp. [65]		100	99.96	99.43	**96.63**	100
	DMBN (Ours)		100	**100**	**99.61**	96.62	100
MeGlass	AdaBN (Agg) [19]	16	98.51	94.63	85.93	72.47	97.63
	AdaBN (Base) [19]		98.87	95.59	88.03	75.81	98.15
	MMD-Linear (Base) [14]		98.81	94.87	87.04	75.65	97.85
	MMD-Gauissian (Base) [14]		98.84	94.88	87.34	76.72	97.78
	Pseudo labeling [17] + Imp. [65]		98.76	95.36	87.58	76.27	97.98
	DMBN (Ours)		**99.03**	**96.15**	**89.43**	**79.11**	**98.43**
Public-IvS	AdaBN (Agg) [19]	16	97.68	93.81	86.87	75.74	93.34
	AdaBN (Base) [19]		97.88	94.48	88.2	76.74	94.06
	MMD-Linear (Base) [14]		97.68	93.83	86.69	74.77	93.02
	MMD-Gauissian (Base) [14]		97.78	93.76	86.94	74.49	92.98
	Pseudo labeling [17] + Imp. [65]		97.92	94.55	88.3	78.25	93.72
	DMBN (Ours)		**98.41**	**96.73**	**92.35**	**84.71**	**96**
WebCaricature	AdaBN (Agg) [19]	16	51.44	38.23	-	-	-
	AdaBN (Base) [19]		57.09	41.4			
	MMD-Linear (Base) [14]		58.71	43.7			
	MMD-Gauissian (Base) [14]		59.94	43.89			
	Pseudo labeling [17] + Imp. [65]		59.45	42.62			
	DMBN (Ours)		**66.29**	**46.13**			

small scale of noised data leads to overfitting. (v) On the most challenging domain of WebCaricature, the performance of DMBN surpasses other UDA competitors over 6.8% at FAR=1% and 2.2% at FAR=0.1%.

E. Comparative Experiments to UDA Competitors with Sufficient Samples

Our method DMBN beats other UDA competitors by a large margin when the target number is only 16 (Section IV-D). To further demonstrate the efficacy of DMBN, we perform a comparison on CASIA NIR-VIS 2.0 across different numbers of target samples. The results in Fig. 4 and Table IV show: (i) Other UDA methods improve the performance benefiting from the increased number of samples, *e.g.*, *MMD-Gaussian (Base)* improves the VR from 80.69% (16 samples) to 83.8% (1,000 samples). (ii) DMBN still performs better than other UDA competitors obviously. For example, DMBN is 4% higher than the best *MMD-Gaussian (Base)* model.

F. Comparison Between UDAL-FR-I and UDAL-FR-II

The UDAL-FR-II benchmark assumes the domain labels of source domains are unavailable, which is more consistent with the real-world scenarios and is thus more challenging than UDAL-FR-I. For UDAL-FR-II, our DMBN incorporates the proposed balanced k-means module to assign pseudo-domain labels within each mini-batch to perform the distribution-shifted task sampling. We compare DMBN under UDAL-FR-I and UDAL-FR-II benchmarks with different numbers of target samples in Table V. The results demonstrate that the overall performance of UDAL-FR-II only drops slightly compared to UDAL-FR-I. For example, on CASIA NIR-VIS 2.0, the VR gaps are less than one percent across different numbers of target samples when FAR=0.01%: 85.81%

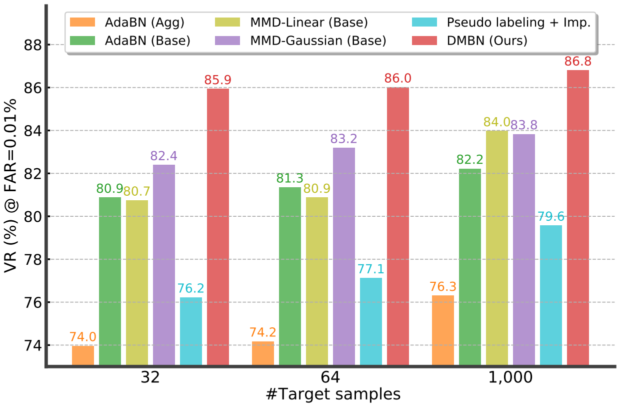


Fig. 4. Comparative results with other UDA competitors on CASIA NIR-VIS 2.0, under the UDAL-FR-I benchmark. From left to right, we show the VRs at FAR=0.01% with different numbers of target samples, *e.g.*, 32, 64 and 1,000 samples. Our DMBN shows better performance than competitors obviously.

vs. 84.9% (16 samples), 85.93% vs. 85.05% (32 samples), 86.01% vs. 85.46% (64 samples) and 86.8% vs. 85.97% (1,000 samples). It is worth noting that DMBN with balanced k-means (UDAL-FR-II) even performs better than DMBN with source domain labels (UDAL-FR-I) on WebCaricature. It is likely that balanced k-means produces more varieties of distribution shifts than source domain labels and such varieties help improve the adaptation on the target domain. The overall results demonstrate that our DMBN is robust even without the domain information, and the balanced k-means is effective for DMBN.

TABLE V
COMPARISON RESULTS OF DMBN BETWEEN UDAL-FR-I AND UDAL-FR-II BENCHMARKS AT FAR=0.01%. DM. INDICATES DOMAIN INFORMATION. THE METHOD DMBN W/ DM. CORRESPONDS TO UDAL-FR-I AND DMBN W/O DM. (k=4) CORRESPONDS TO UDAL-FR-II. k IS THE CLUSTER NUMBER OF BALANCED k-MEANS AND SET 4 HERE.

Dataset / Domain	**Method**	**#Target samples**			
		16	**32**	**64**	**1,000**
		VR(%)@FAR=0.01%			
CASIA NIR-VIS 2.0	DMBN w/ dm. DMBN w/o dm. (k=4)	**85.81** 84.9	**85.93** 85.05	**86.01** 85.46	**86.8** 85.97
HFB	DMBN w/ dm. DMBN w/o dm. (k=4)	**95.38** 95.15	**96.45** 95.67	**96.63** 95.78	**96.72** 96.14
Oulu-CASIA NIR-VIS	DMBN w/ dm. DMBN w/o dm. (k=4)	73.7 **73.73**	**74.46** 73.98	76.63 **76.78**	**76.93** 76.68
MultiPIE	DMBN w/ dm. DMBN w/o dm. (k=4)	**99.43** 99.04	**99.51** 99.01	**99.6** 99.1	**99.61** 99.08
MeGlass	DMBN w/ dm. DMBN w/o dm. (k=4)	**89.43** 88.97	**90.28** 89.8	**90.48** 90.09	**90.81** 90.24
Public-IvS	DMBN w/ dm. DMBN w/o dm. (k=4)	**92.35** 91.89	**92.8** 91.9	**93.02** 92.32	**93.15** 92.59
		VR(%)@FAR=1%			
WebCaricature	DMBN w/ dm. DMBN w/o dm. (k=4)	66.29 **67.95**	66.23 **67.32**	66.64 **68.32**	67.55 **69.32**

G. Comparative Experiments With Supervised Competitors

We compare our *unsupervised* DMBN to other *supervised* competitors on Oulu-CASIA NIR-VIS, which has a larger illumination and expression gap to source domains. The supervised competitors, *e.g.*, WCNN [66], DVR [67], use labeled samples from the target domain for training. The results are shown in Table VI. We find our results are competitive to supervised methods. Our performance even surpasses most of other supervised methods, *e.g.*, Light CNN [68], IDR [69], DVR [67], ADFL [70] and WCNN [66], which use all 5,760 target labeled samples for training. The results show that our unsupervised DMBN is a potential alternative for supervised methods.

TABLE VI
COMPARATIVE RESULTS WITH OTHER SUPERVISED METHODS ON OULU-CASIA NIR-VIS, UNDER THE UDAL-FR-II BENCHMARK. DOMAIN LABELS OF TRAINING DOMAINS ARE UNAVAILABLE IN UDAL-FR-II. k IS THE CLUSTER SIZE AND IS SET 4 HERE. #TS. INDICATES THE NUMBER OF TARGET SAMPLES. NOTE THAT OTHER COMPETITORS ARE ALL SUPERVISED METHODS AND ARE TRAINED WITH ABOUT 5,760 LABELED SAMPLES. LR. INDICATES LOW-RANK.

Method	**#Ts.**	**VR (%)**		**Rank1 (%)**
		FAR=1%	**FAR=0.1%**	
Agg Base	-	95.09 96.82	82.54 86.4	100 100
Light CNN [68] IDR [69] ADFL [70] WCNN [66] WCN + lr. [66] DVR [67] DVG [71]	5,760	92.4 73.4 83 75 81.5 97.2 **98.5**	65.1 46.2 60.7 50.9 54.6 84.9 **92.9**	96.7 94.3 95.5 96.4 98 100 **100**
DMBN (k=4)	16 32 64 1,000	96.94 96.78 97.01 **97.5**	85.33 86.32 88.02 **88.84**	100 100 100 **100**

H. Ablation Study

Domain-shuffling. During training, we randomly shuffle the meta-train batches across tasks to enlarge the distribution shift between meta-train and meta-test. To evaluate its contribution, we perform ablation experiments on CASAI NIR-VIS 2.0 with five settings: (i) Randomly sampling without using source domain labels or estimated domain labels by balanced k-means. It means the samples are all randomly sampled. (ii) Using estimated domain labels by balanced k-means without domain shuffling. (iii) Using source domain labels without domain shuffling. (iv) Using estimated domain labels by balanced k-means with domain shuffling. (v) Using source domain labels with domain shuffling. The results in Fig. 5 show: with different numbers of target samples as input, domain shuffling improves the performance with source domain labels or estimated domain labels by balanced k-means. For example, when the number of target samples is 32, the verification rate at FAR=0.001% improves 2.7% with source domain labels and 3% with estimated domain labels by balanced k-means. Besides, the performance of domain shuffling with source or estimated domain labels surpasses the random sampling.

The impact of γ. In Eqn. 5, the hyper-parameter γ weights the meta-train and meta-test losses. To analyze the impact of γ,

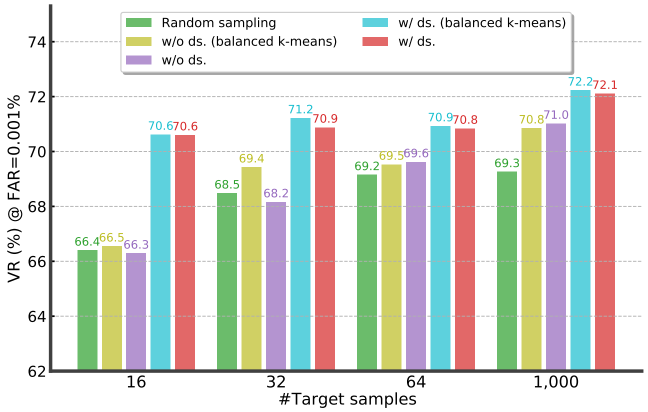


Fig. 5. The ablation results of domain shuffling on CASIA NIR-VIS 2.0 at the low FAR=0.001% with different target samples. ds. is the domain shuffling operation. Five settings are compared: (i) random sampling means sampling without using source or estimated domain labels. (ii) w/o ds. (balanced k-means) means not using domain shuffling with the estimated domain labels by balanced k-means. (iii) w/o ds. means not using the domain shuffling. (iv) w/ ds. (balanced k-means) indicates the domain labels are estimated by balanced k-means. (v) w/ ds. uses the source domain labels for domain shuffling.

we conduct ablative experiments in Fig. 6. The results indicate that the gradients from the meta-train and meta-test should be roughly equally weighted in optimization. A proper value, *e.g.*, 0.4, 0.5, or 0.6, gives satisfied results.

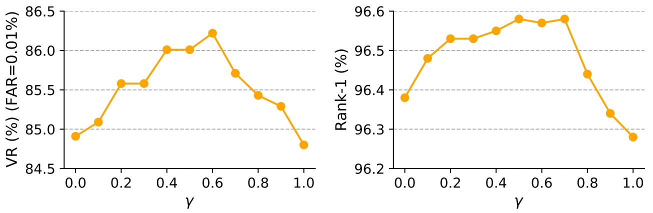


Fig. 6. The ablation results on CASIA NIR-VIS 2.0, under the UDAL-FR-I benchmark, with different γ. The number of target samples is 64.

Comparison between vanilla and balanced k-means. When domain labels are unavailable in UDAL-FR-II, DMBN adopts the balanced k-means to assign pseudo-domain labels within batches for training. Compared to the vanilla k-means, the balanced k-means can produce k clusters with the same size. As shown in Fig 8, the vanilla k-means tends to group the "Asian" and "Caucasian" domains together, making the clustering results unbalanced. The balanced k-means shows a good separation and balances the outputted cluster size simultaneously. Besides, we compare the quantitative results in Table VII. The results show that the balanced k-means performs better than the vanilla one for DMBN. For example, when FAR=0.01% and the number of target samples is 32, the VR of the balanced k-means is 85.05%, surpassing the vanilla one 82.55% by a large margin.

Balanced k-means on unbalanced source domains. To further verify the effectiveness of balanced k-means with the unbalanced distribution, we conduct comparative experiments on unbalanced source domains $\mathcal{D}_S^{'}$ in Fig. 7. $\mathcal{D}_S^{'}$ is built by adjusting the race distribution of source datasets to be consistent with the distribution of Ms-Celeb-1M given by [55]. After the adjustment, the training ids/images of each dataset from the unbalanced $\mathcal{D}_S^{'}$ are: 2,958 ids/10,185 images (Caucasian), 256 ids/1,867 images (Asian), 562 ids/3,748 images (African) and 101 ids/666 images (Indian). Obviously, Caucasians dominate the distribution of $\mathcal{D}_S^{'}$. In training, the domain information is unknown following the UDAL-FR-II protocol. The comparative results in Fig. 7 show that the balanced k-means outperforms vanilla k-means even on unbalanced source domains.

TABLE VII

THE COMPARATIVE RESULTS ON CASIA NIR-VIS 2.0 BETWEEN THE VANILLA K-MEANS AND BALANCED K-MEANS. #TS. INDICATES THE NUMBER OF AVAILABLE SAMPLES IN THE TARGET DOMAIN.

#Ts.	k-means	VR (%) @ FAR=			Rank1 (%)
		0.1%	0.01%	0.001%	
16	Vanilla Balanced	93.86 **94.93**	82.13 **84.9**	66.86 **70.62**	95.68 **96.29**
32	Vanilla Balanced	93.94 **94.92**	82.55 **85.05**	66.03 **71.22**	95.66 **96.32**
64	Vanilla Balanced	94.33 **94.93**	83.84 **85.46**	67.72 **70.93**	95.86 **96.23**
1,000	Vanilla Balanced	94.5 **95.18**	84.12 **85.97**	69.78 **72.23**	96.08 **96.4**

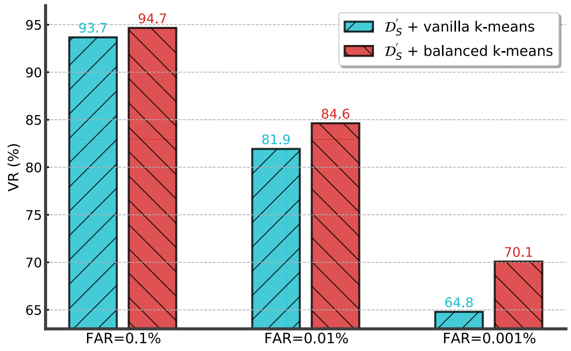


Fig. 7. The comparative results of CASIA NIR-VIS 2.0 on unbalanced source domains $\mathcal{D}_S^{'}$. The vanilla k-means and balanced k-means are compared.

Impact of k. k is a hyper-parameter of the balanced k-means, which represents the desired number of clusters. Once set, k is fixed during training. The ablation results are shown in Fig. 9. When k becomes larger, *e.g.*, greater than 6, the performance drops rapidly on CASIA NIR-VIS 2.0. This is possibly because the distribution shifts across clustered domains get smaller when k gets larger. In Fig. 9, when considering the overall performance (the mean performance of all FARs), we set k=4.

I. Discussions

Sampling of target samples. In the adaptation phase, the sampling of target samples is completely random, which may sample an unbalanced distribution from the target domain. In comparison, by-individual sampling is balanced to some degree. For example, when sampling 1,000 samples from CASIA

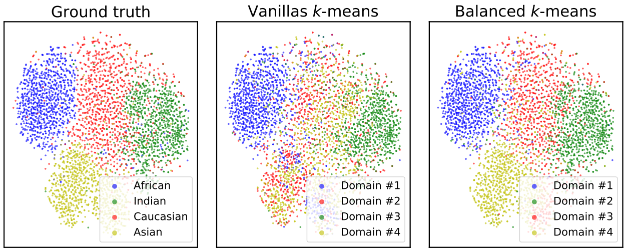


Fig. 8. The t-SNE visualizations of the clustering results by the vanilla k-means and balanced k-means. The left one is with groudtruth labels, the middle one is vanilla k-means and the right is balanced k-means.

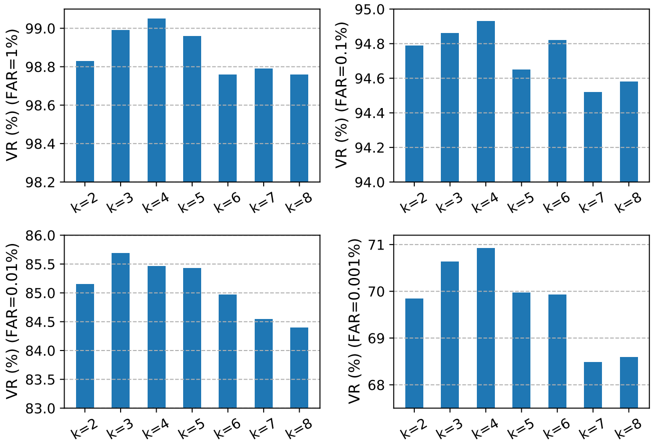


Fig. 9. Ablation results on CASIA NIR-VIS 2.0, with different k for balanced k-means, at FAR=1%, 0.1%, 0.01% and 0.001%, under the UDAL-FR-II benchmark. The number of target samples is 64.

NIR-VIS 2.0, the by-individual sampling evenly samples about three samples from each individual. The comparative results in Fig. 10 show that the balanced by-individual sampling is slightly better than by-sample. In other words, our DMBN is robust to unbalanced sampling.

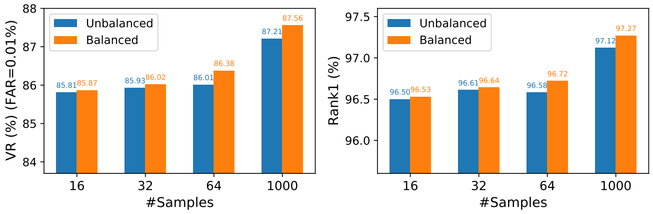


Fig. 10. The comparative results on CASIA NIR-VIS 2.0, under the UDAL-FR-I benchmark, with different sampling of the target samples.

Updating momentum of $\mathcal{BN}$. In the adaptation phase, we directly use target samples to estimate the BN statistics $\mathcal{BN}$, ignoring the original $\mathcal{BN}$ of the trained model. In other words, the momentum is equivalent to 0 in our setting. Fig. 11 compares the effect of different momentums on the verification rate (VR) of HFB. We can observe that when the momentum gets larger, the performance gets worse, and the zero momentum performs best. These results indicate that the $\mathcal{BN}$ of the original model hampers the adaptation.

Computation cost of balanced k-means. The time complexity of vanilla k-means is $O(Tdn^2)$ if considering the distance matrix calculation, where T is the number of iteration, d is the feature dimension and n is the number of input samples. In comparison, the time complexity of balanced k-means is also $O(Tdn^2)$. The extra computation complexity brought by balanced k-means is $O(Tn^2)$ in the adjustment phase, which has the same magnitude as the overall complexity. Theoretically, our balanced k-means imposes only a small computation overhead over vanilla k-means. Comparing vanilla and balanced k-means when the whole batch size is set 1,024, the time is 0.156s vs. 0.185s, respectively. The practical results also show the computation overhead of balanced k-means is small.

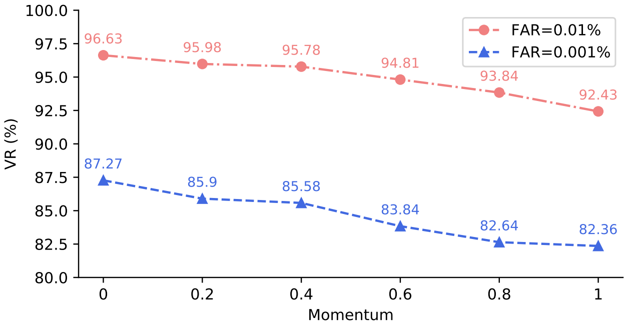


Fig. 11. The performance on HFB, under the UDAL-FR-I benchmark, with different momentums to update $\mathcal{BN}$.

Adaptation efficiency. To perform adaptation, the model learned by DMBN only needs to update the BN with limited samples from the target domain. The speed of adaptation is thus very fast. For example, the adaptation only spends 7.67ms (217ms) on GPU or 0.21s (16s) on CPU with 16 (1,000) samples as input. In comparison, other UDA methods, *e.g.*, MMD or pseudo labeling based methods, need gradient updating and take up at least several minutes or hours to perform adaptation.

TABLE VIII
COMPARISON OF DG AND UDA METHODS ON CASIA NIR-VIS 2.0.

Setting	Method	VR (%) @ FAR=			Rank1 (%)
		0.1%	**0.01%**	**0.001%**	
DG	MLDG [72]	90.44	69.32	-	93.56
	MFR [51]	95.97	81.92	-	96.92
UDA	**DMBN (Ours)**	**96.19**	**86.8**	**72.11**	**97.12**

Comparison between UDA and DG methods. As a highly related direction to unsupervised domain adaptation (UDA), domain generalization (DG) assumes that the data of target domain is inaccessible during training. The UDA and DG methods are rarely compared before. We compare our unsupervised DMBN with the other two DG methods on CASIA NIR-VIS 2.0 in Table VIII. From the results, we can see that DMBN outperforms DG methods, especially at the low FAR=0.01%.

V. Conclusion

In this paper, we highlight the challenging problem of Unsupervised Domain Adaptation for Face Recognition with Limited samples (UDAL-FR), which usually exists in real-world scenarios. To address it, we propose a novel meta-learning based framework by decomposing the model into the weight parameters θ and the BN statistics $\mathcal{BN}$, named Decomposed Meta Batch Normalization (DMBN). DMBN trains the network such that domain-invariant information is prone to store in θ and domain-specific knowledge tends to be represented by $\mathcal{BN}$. Once trained, the model can fast adapt to the target domain via only updating $\mathcal{BN}$ with limited samples from the target domain. Extensive experiments on two newly defined UDAL-FR benchmarks validate the efficacy of our proposed DMBN. We believe the UDAL-FR problem is of great importance for real-world face recognition applications, and hope our method paves an avenue for future works.

References

[1] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1701–1708.

[2] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1891–1898.

[3] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[4] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 212–220.

[5] J. Deng, J. Guo, X. Niannan, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4690–4699.

[6] J. Deng, J. Guo, T. Liu, M. Gong, and S. Zafeiriou, "Sub-center arcface: Boosting face recognition by large-scale noisy web faces," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 741–757.

[7] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," *Technical Report*, pp. 07–49, Oct. 2007.

[8] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 529–534.

[9] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney *et al.*, "Iarpa janus benchmark-c: Face dataset and protocol," in *Proc. Int. Conf. Biometrics (ICB)*, Feb. 2018, pp. 158–165.

[10] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4873–4882.

[11] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch." [Online]. Available: http://arxiv.org/abs/1411.7923

[12] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *Proc. 13th IEEE Int. Conf. Autom. Face Gesture Recognit (FG)*, May. 2018, pp. 67–74.

[13] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, May. 2016, pp. 87–102.

[14] Z. Luo, J. Hu, W. Deng, and H. Shen, "Deep unsupervised domain adaptation for face recognition," in *Proc. 13th IEEE Int. Conf. Autom. Face Gesture Recognit (FG)*, May. 2018, pp. 453–457.

[15] K. Sohn, S. Liu, G. Zhong, X. Yu, M.-H. Yang, and M. Chandraker, "Unsupervised domain adaptation for face recognition in unlabeled videos," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3210–3218.

[16] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2015, pp. 97–105.

[17] Q. Zhang, Z. Lei, and S. Z. Li, "Neighborhood-aware attention network for semi-supervised face recognition," in *Int. Joint Conf. on Neural Networks (IJCNN)*, Jul. 2020, pp. 1–8.

[18] M. Wang and W. Deng, "Deep face recognition with clustering based domain adaptation," *Neurocomputing*, vol. 393, pp. 1–14, May. 2020.

[19] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu, "Adaptive batch normalization for practical domain adaptation," *Pattern Recognit. (PR)*, vol. 80, pp. 109–117, Feb. 2018.

[20] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Aug. 2017, pp. 1126–1135.

[21] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2014, pp. 1988–1996.

[22] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 499–515.

[23] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks." in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jun. 2016, pp. 507–516.

[24] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 5265–5274.

[25] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Process. Lett.*, vol. 25, no. 7, pp. 926–930, Wed. 2018.

[26] X. Liang, X. Wang, Z. Lei, S. Liao, and S. Z. Li, "Soft-margin softmax for deep classification," in *Proc. Int. Conf. on Neural Inf. (ICONIP)*, Nov. 2017, pp. 413–421.

[27] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014. [Online]. Available: http://arxiv.org/abs/1412.3474

[28] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Aug. 2017, pp. 2208–2217.

[29] M. Long, H. Zhu, J. Wang, and M. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Jan. 2016, pp. 136–144.

[30] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2272–2281.

[31] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2015, pp. 1180–1189.

[32] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Feb. 2018, pp. 3934–3941.

[33] W. Zhang, W. Ouyang, W. Li, and D. Xu, "Collaborative and adversarial network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 3801–3809.

[34] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4068–4076.

[35] S. Motiian, Q. Jones, S. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2017, pp. 6670–6680.

[36] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, pp. 59:1–59:35, Jul. 2016.

[37] Z. Cao, M. Long, J. Wang, and M. I. Jordan, "Partial transfer learning with selective adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 2724–2732.

[38] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, "Importance weighted adversarial nets for partial domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8156–8164.

[39] P. Panareda Busto and J. Gall, "Open set domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 754–763.

[40] K. Saito, S. Yamamoto, Y. Ushiku, and T. Harada, "Open set domain adaptation by backpropagation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 156–171.

[41] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "One-shot learning with memory-augmented neural networks." [Online]. Available: http://arxiv.org/abs/1605.06065

[42] T. Munkhdalai and H. Yu, "Meta networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Aug. 2017, pp. 2554–2563.

[43] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. Int. Conf. Mach. Learn. Workshops (ICMLW)*, Jul. 2015.

[44] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2017, pp. 4077–4087.

[45] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2016, pp. 3630–3638.

[46] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 1199–1208.

[47] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," 2018. [Online]. Available: http://arxiv.org/abs/1803.02999

[48] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 403–412.

[49] A. Rajeswaran, C. Finn, S. Kakade, and S. Levine, "Meta-learning with implicit gradients," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2019, pp. 113–124.

[50] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2015, pp. 448–456.

[51] J. Guo, X. Zhu, C. Zhao, D. Cao, Z. Lei, and S. Z. Li, "Learning meta face recognition in unseen domains," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6163–6172.

[52] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. Sym. on Disc. Algo. (SODA)*, Jan. 2007, pp. 1027–1035.

[53] D. Cao, X. Zhu, X. Huang, J. Guo, and Z. Lei, "Domain balancing: Face recognition on long-tailed domains," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5671–5679.

[54] T. F. Chan, G. H. Golub, and R. J. LeVeque, "Updating formulae and a pairwise algorithm for computing sample variances," in *COMPSTAT*, 1982, pp. 30–41.

[55] M. Wang, W. Deng, J. Hu, J. Peng, X. Tao, and Y. Huang, "Racial faces in-the-wild: Reducing racial bias by information maximization adaptation network," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 692–702.

[56] S. Z. Li, D. Yi, Z. Lei, and S. Liao, "The casia nir-vis 2.0 face database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2013, pp. 348–353.

[57] S. Z. Li, Z. Lei, and M. Ao, "The hfb face database for heterogeneous face biometrics research," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2009, pp. 1–8.

[58] Z. Lei, M. Pietikäinen, and S. Z. Li, "Learning discriminant face descriptor," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 289–302, Jun. 2013.

[59] J. Chen, D. Yi, J. Yang, G. Zhao, S. Z. Li, and M. Pietikainen, "Learning mappings for face synthesis from near infrared to visual light images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 156–163.

[60] M. Shao and Y. Fu, "Cross-modality feature learning through generic hierarchical hyperlingual-words," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 2, pp. 451–463, Jan. 2016.

[61] J. Guo, X. Zhu, Z. Lei, and S. Z. Li, "Face synthesis for eyeglass-robust face recognition," in *Chinese Conf. Bio. Recognit. (CCBR)*, Aug. 2018, pp. 275–284.

[62] X. Zhu, H. Liu, Z. Lei, H. Shi, F. Yang, D. Yi, G. Qi, and S. Z. Li, "Large-scale bisample learning on id versus spot face recognition," *Int. J. Comput. Vis.*, vol. 127, no. 6-7, pp. 684–700, Feb. 2019.

[63] J. Huo, W. Li, Y. Shi, Y. Gao, and H. Yin, "Webcaricature: a benchmark for caricature recognition," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Sep. 2017, p. 223.

[64] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. Conf. Neural Inf. Process. Syst. Workshop (NeurIPS Workshop)*, Dec. 2017.

[65] H. Qi, M. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 5822–5830.

[66] R. He, X. Wu, Z. Sun, and T. Tan, "Wasserstein cnn: Learning invariant features for nir-vis face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1761–1773, Jul. 2019.

[67] X. Wu, H. Huang, V. M. Patel, R. He, and Z. Sun, "Disentangled variational representation for heterogeneous face recognition," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Jan. 2019, pp. 9005–9012.

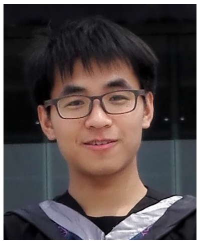
[68] R. He, J. Cao, L. Song, Z. Sun, and T. Tan, "Adversarial cross-spectral face completion for nir-vis heterogeneous face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1025–1037, Dec. 2019.

[69] R. He, X. Wu, Z. Sun, and T. Tan, "Learning invariant deep representation for nir-vis face recognition," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Feb. 2017, pp. 2000–2006.

[70] L. Song, M. Zhang, X. Wu, and R. He, "Adversarial discriminative heterogeneous face recognition," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Feg. 2018, pp. 7355–7362.

[71] C. Fu, X. Wu, Y. Hu, H. Huang, and R. He, "Dual variational generation for low shot heterogeneous face recognition," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2019, pp. 2670–2679.

[72] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Feb. 2018, pp. 3490–3497.



Jianzhu Guo received the B.E. degree from the School of Transportation, Southeast University (SEU), Nanjing, China, in 2016. He is currently pursuing the Ph.D. degree with the Institute of Automation, Chinese Academy of Science (CASIA). His main research interests include face recognition, 3D face, face anti-spoofing, face analysis and meta learning, deep learning.

Xiangyu Zhu received the BS degree in Sichuan University (SCU) in 2012, and the PhD degree from Institute of Automation, Chinese Academy of Sciences, in 2017, where he is currently an associate professor. His research interests include pattern recognition and computer vision, in particular, image processing, 3D face model, face alignment and face recognition.

Zhen Lei received the B.S. degree in automation from the University of Science and Technology of China, in 2005, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2010, where he is currently a Professor. He has published over 100 papers in international journals and conferences. His research interests are in computer vision, pattern recognition, image processing, and face recognition in particular. He served as an Area Chair of the International Joint Conference on Biometrics in 2014, the IAPR/IEEE International Conference on Biometric in 2015, 2016, 2018, and the IEEE International Conference on Automatic Face and Gesture Recognition in 2015.

Stan Z. Li received his B.Eng from Hunan University, China, M.Eng from National University of Defense Technology, China, and PhD degree from Surrey University, UK. He is currently a professor and the director of Center for Biometrics and Security Research (CBSR), Institute of Automation, Chinese Academy of Sciences (CASIA). He worked at Microsoft Research Asia as a researcher from 2000 to 2004. Prior to that, he was an associate professor at Nanyang Technological University, Singapore. He was elevated to IEEE Fellow for his contributions to the fields of face recognition, pattern recognition and computer vision. His research interest includes pattern recognition and machine learning, image and vision processing, face recognition, biometrics, and intelligent video surveillance. He has published over 200 papers in international journals and conferences, and authored and edited 8 books. He served as a program co-chair for the International Conference on Biometrics 2007 and 2009, and has been involved in organizing other international conferences and workshops in the fields of his research interest.